Build an ASP.NET Core Web App with F#? You must be having a **Giraffe!**

GIRAFFE

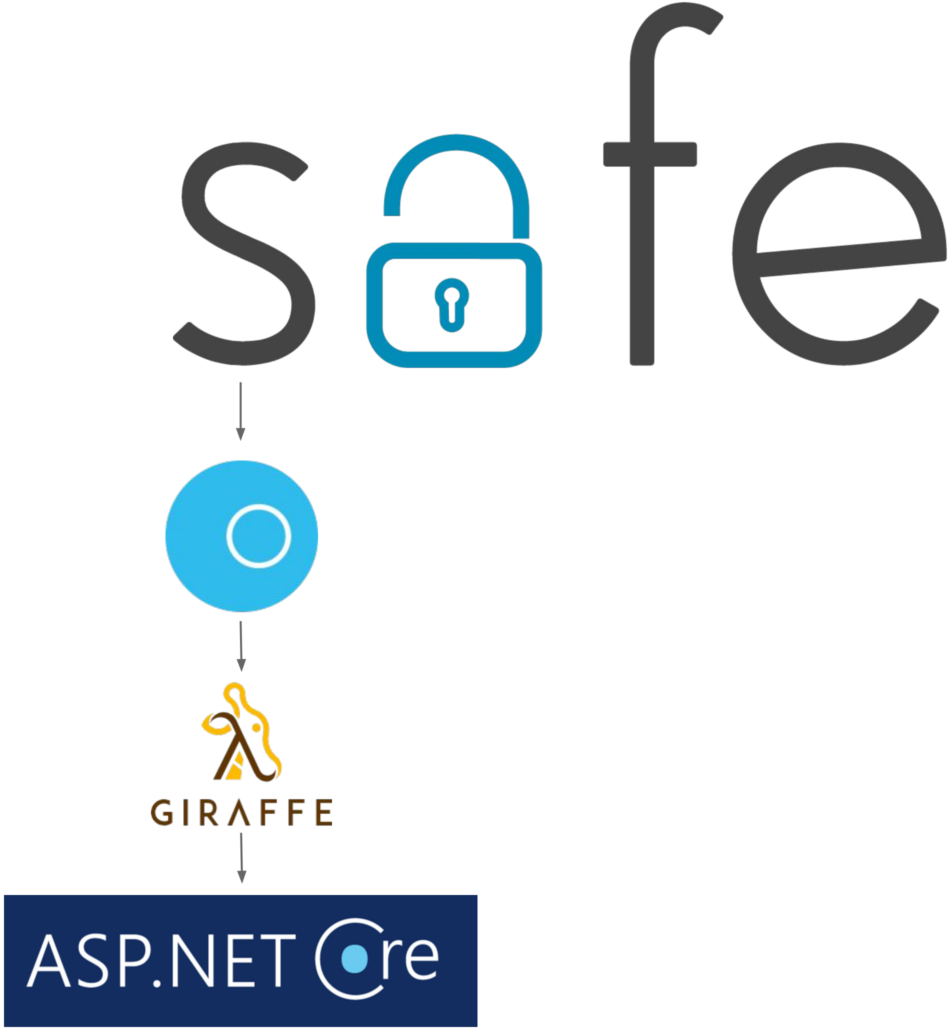
What I’ll cover

- What is Giraffe
- Where does it fit in?
- How does it work?

1

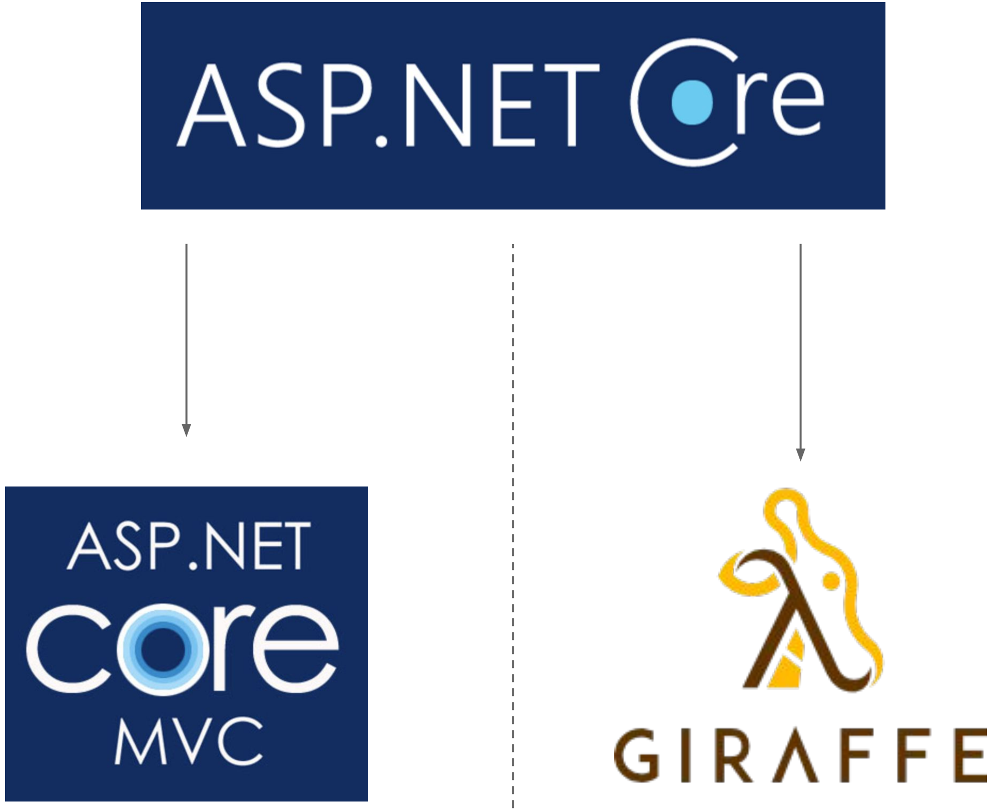
What is it?

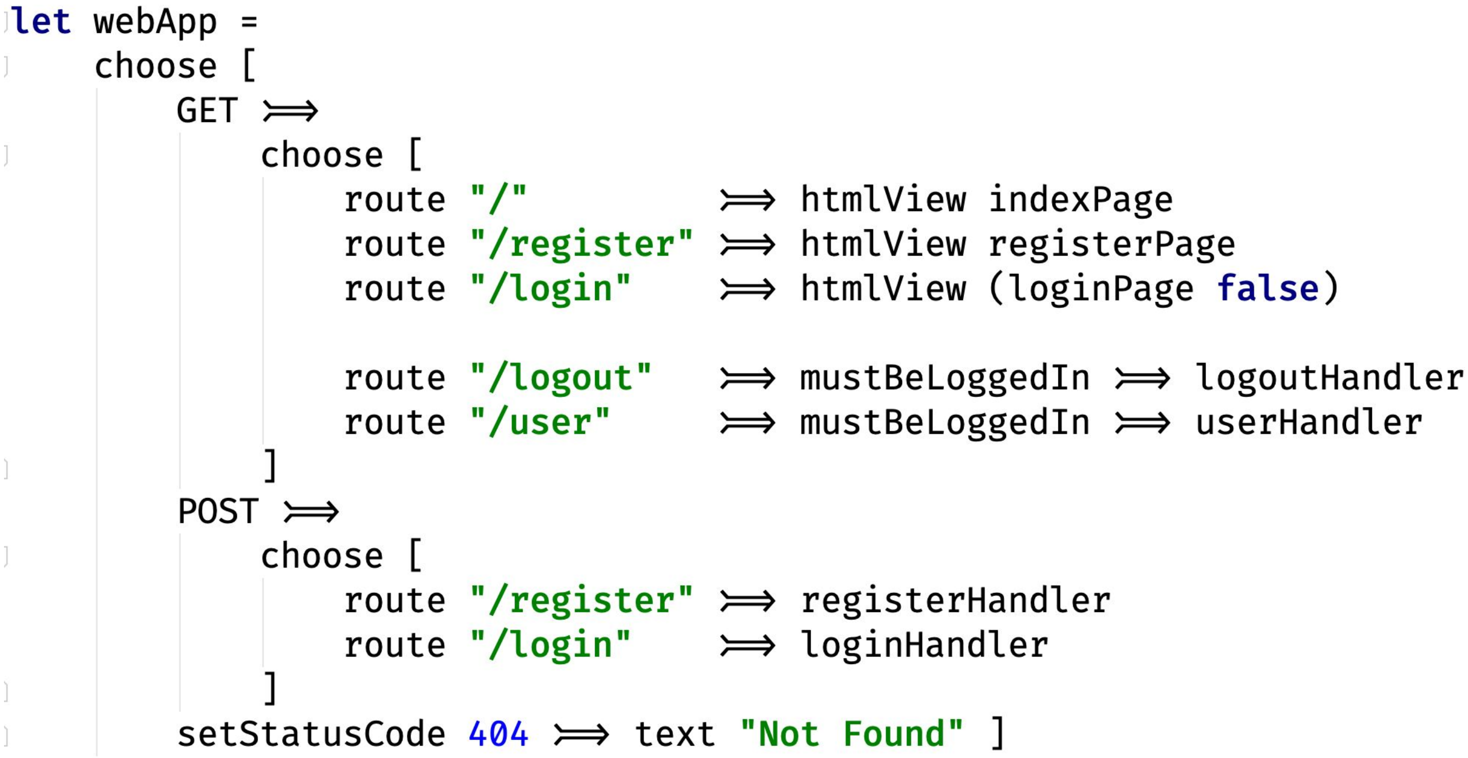
A *functional* ASP.NET Core micro *web framework for building rich web applications.*


safe
GIRAFFE
ASP.NET Core

Idiomatic C# 😞

Idiomatic F# 😎





```
let webApp =
    choose [
        GET >=>
            choose [
                route "/"          >=> htmlView indexPage
                route "/register"  >=> htmlView registerPage
                route "/login"     >=> htmlView (loginPage false)

                route "/logout"    >=> mustBeLoggedIn >=> logoutHandler
                route "/user"      >=> mustBeLoggedIn >=> userHandler
            ]
        POST >=>
            choose [
                route "/register"  >=> registerHandler
                route "/login"     >=> loginHandler
            ]
        setStatusCode 404 >=> text "Not Found" ]
```

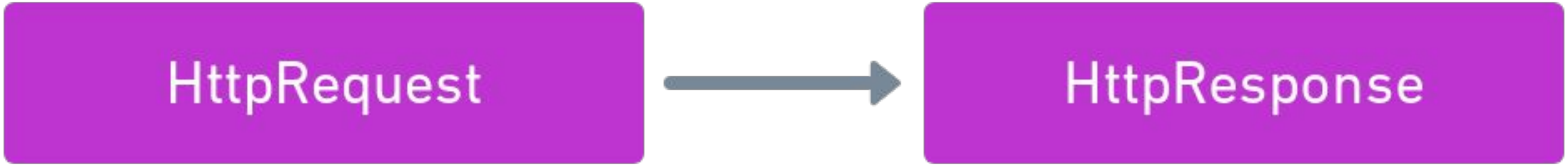
An example Giraffe app (*source*)

1

How does it work?

Let’s model it!

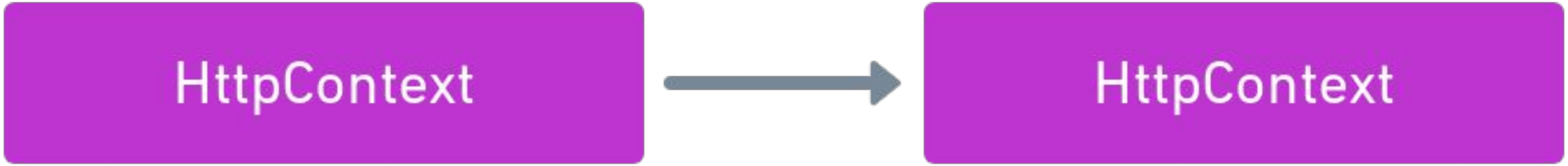
```
type HttpServer =
```



HttpRequest → HttpResponse

Let’s model it!

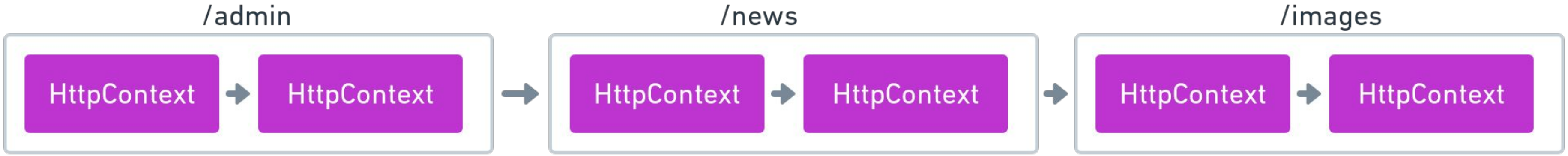
type HttpHandler =



HttpContext

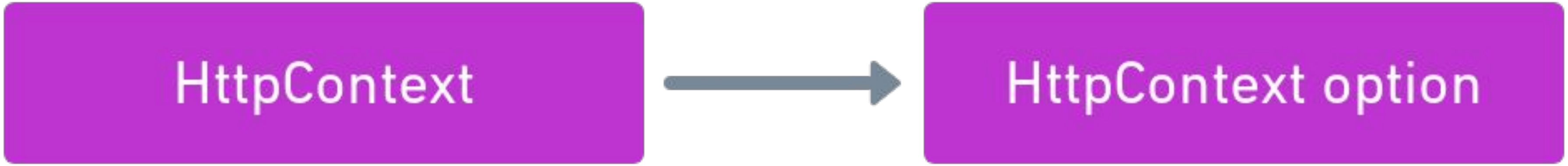
HttpContext

Let’s model it!



Let’s model it!

type HttpHandler =

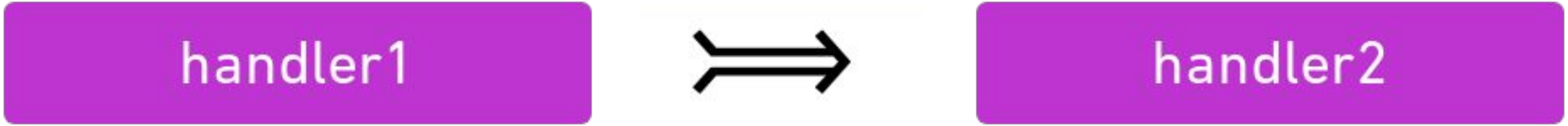


HttpContext

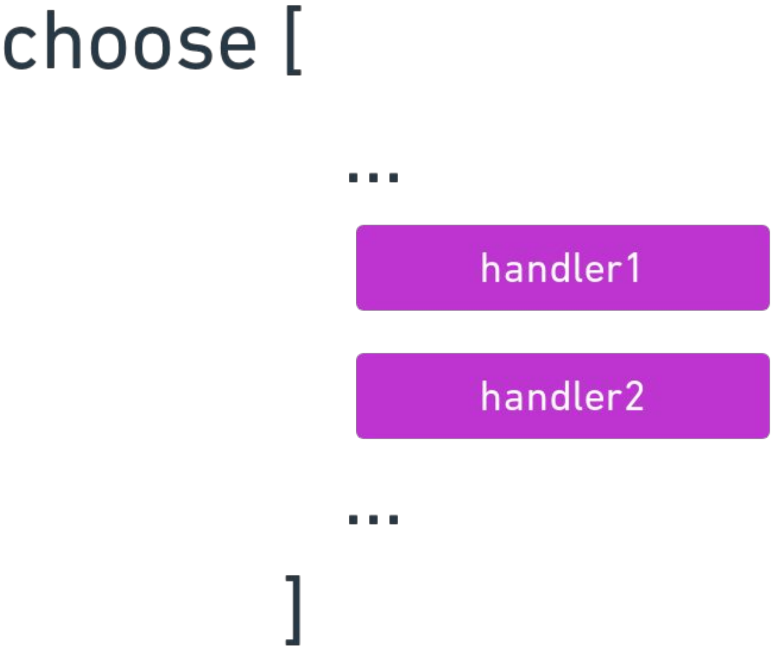
HttpContext option

Let’s model it!

(compose)



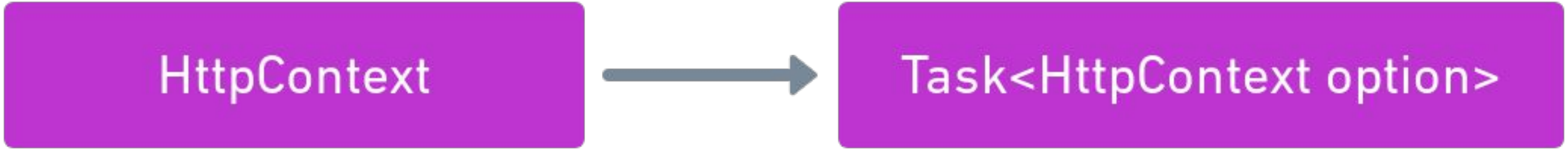
Let’s model it!



```
choose [
        ...
         handler1
         handler2
        ...
       ]
```

Let’s model it!

```
type HttpHandler =
```



HttpContext → Task<HttpContext option>

Let’s model it!

But it could be better…

https://medium.com/@gerardtoconnor/carry-on-continuation-over-binding-pipelines-for-functional-web-58bd7e6ea009

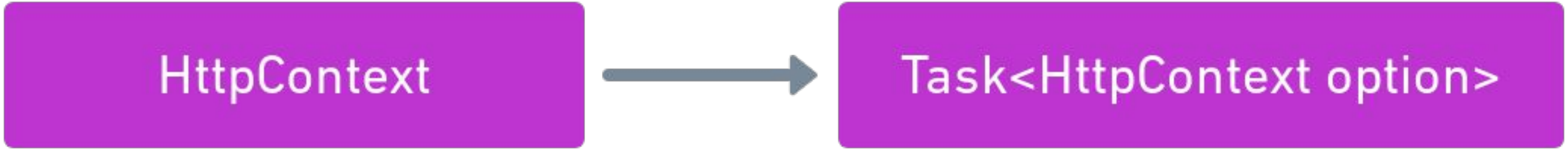
Carry On! … Continuation over binding pipelines for functional web

Gerard

Jul 13, 2017 · 6 min read

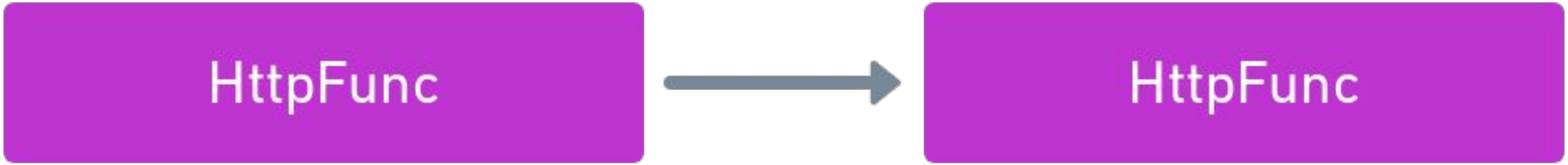
Let's model it!

type HttpFunc =



HttpContext → Task<HttpContext option>

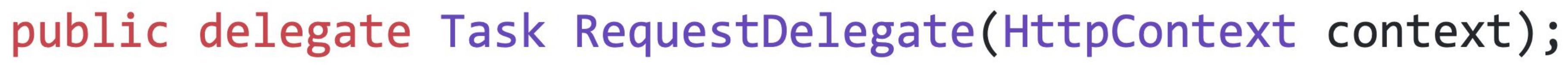
type HttpHandler =



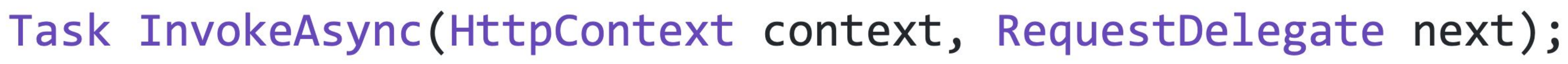
HttpFunc → HttpFunc

Let’s model it!

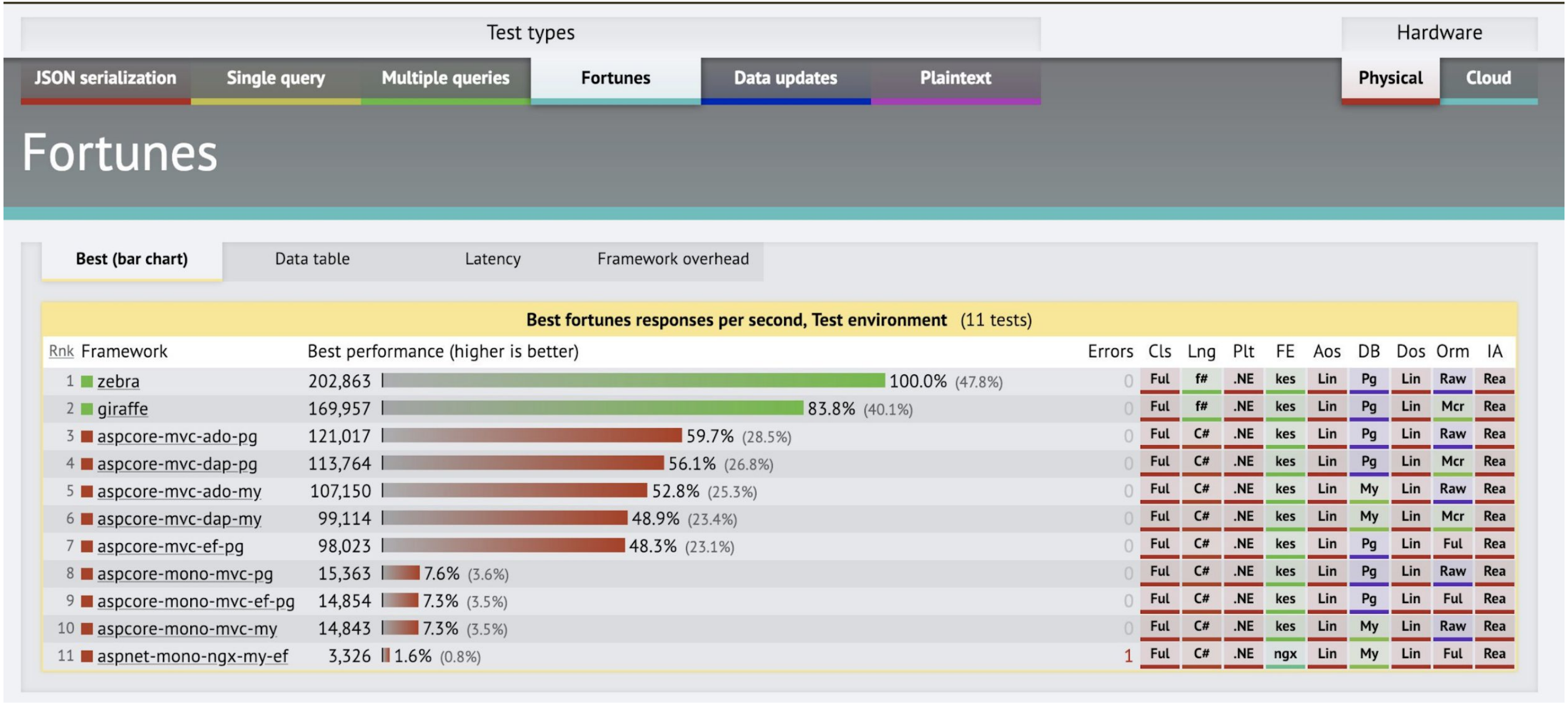
Familiar… ASP.NET Core Middleware



```
public delegate Task RequestDelegate(HttpContext context);
```



```
Task InvokeAsync(HttpContext context, RequestDelegate next);
```



Fortunes

Best fortunes responses per second, Test environment (11 tests)

Rnk	Framework	Best performance (higher is better)	Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	🟩 zebra	202,863 100.0% (47.8%)	0	Ful	f#	.NE	kes	Lin	Pg	Lin	Raw	Rea
2	🟩 giraffe	169,957 83.8% (40.1%)	0	Ful	f#	.NE	kes	Lin	Pg	Lin	Mcr	Rea
3	🟥 aspcore-mvc-ado-pg	121,017 59.7% (28.5%)	0	Ful	C#	.NE	kes	Lin	Pg	Lin	Raw	Rea
4	🟥 aspcore-mvc-dap-pg	113,764 56.1% (26.8%)	0	Ful	C#	.NE	kes	Lin	Pg	Lin	Mcr	Rea
5	🟥 aspcore-mvc-ado-my	107,150 52.8% (25.3%)	0	Ful	C#	.NE	kes	Lin	My	Lin	Raw	Rea
6	🟥 aspcore-mvc-dap-my	99,114 48.9% (23.4%)	0	Ful	C#	.NE	kes	Lin	My	Lin	Mcr	Rea
7	🟥 aspcore-mvc-ef-pg	98,023 48.3% (23.1%)	0	Ful	C#	.NE	kes	Lin	Pg	Lin	Ful	Rea
8	🟥 aspcore-mono-mvc-pg	15,363 7.6% (3.6%)	0	Ful	C#	.NE	kes	Lin	Pg	Lin	Raw	Rea
9	🟥 aspcore-mono-mvc-ef-pg	14,854 7.3% (3.5%)	0	Ful	C#	.NE	kes	Lin	Pg	Lin	Ful	Rea
10	🟥 aspcore-mono-mvc-my	14,843 7.3% (3.5%)	0	Ful	C#	.NE	kes	Lin	My	Lin	Raw	Rea
11	🟥 aspnet-mono-ngx-my-ef	3,326 1.6% (0.8%)	1	Ful	C#	.NE	ngx	Lin	My	Lin	Ful	Rea

Performance!

It’s a fullstack framework!

- View engines
  - DotLiquid
  - Razor
  - GiraffeViewEngine (HTML DSL)
- Much more

Resources

- Documentation
- Dustin Moris Gorski – Functional ASP.NET Core
- Zaid Ajaj Giraffe helpers

We use it!

- sse.co.uk

Summary

- Giraffe is great!
- We are spoilt for choice with web frameworks in F#

Thanks!

- ***Find me at:***

stu.dev